Modelización del generador auto-shrinking mediante autómatas celulares

A. Fúster-Sabater[1], M.E. Pazo-Robles[2] y P. Caballero-Gil[3]

***Resumen*—En este trabajo se ha desarrollado un modelo lineal muy simple basado en Autómatas Celulares que genera exactamente la misma secuencia de salida que el generador Auto-Shrinking, un conocido generador de secuencia cifrante utilizado en el procedimiento de cifrado en flujo. El proceso de obtención del modelo lineal a partir de los parámetros del generador Auto-Shrinking es inmediato. La implementación del modelo ya linealizado puede llevarse a la práctica mediante lógica FPGA. La linealidad y las propiedades de simetría que aparecen en este modelo celular pueden utilizarse convenientemente para el análisis y/ó criptoanálisis de este tipo de generador criptográfico.**

***Palabras clave*—Autómata celular, criptografía, generador auto-shrinking, modelo lineal, secuencia cifrante.**

I. Introducción

La criptografía de clave secreta se divide en dos grandes apartados: procedimientos de cifrado en flujo y procedimientos de cifrado en bloque. Los sistemas de cifrado en flujo son los más rápidos dentro de los métodos de cifrado, de ahí que en la actualidad se utilicen en numerosas aplicaciones prácticas. Prueba de ello son, por ejemplo, los algoritmos A5 (en su doble versión A5/1 y A5/2) que se utilizan en telefonía móvil GSM [1], el algoritmo E0 usado en especificaciones de Bluetooth [2] ó el algoritmo RC4 utilizado en Microsoft Word y Excel [3].

Un sistema de cifrado en flujo está compuesto por un algoritmo o generador de secuencia cifrante (conocido públicamente) y una clave de cifrado (conocida únicamente por los dos comunicantes). La clave corresponde normalmente al estado inicial del generador de secuencia. En el momento de su inicialización el generador toma la clave como semilla, posteriormente se ejecuta el algoritmo mediante un clock a la velocidad que necesite la aplicación y así se genera la secuencia cifrante.

Para cifrar, el emisor realiza una operación XOR, bit a bit, entre la secuencia cifrante y el texto claro, dando origen al texto cifrado que es el que se va a enviar por el canal de información. Para descifrar, el receptor genera la misma secuencia cifrante que suma bit a bit con el texto cifrado recibido y recupera así el texto claro original.

Muchos algoritmos de cifrado en flujo están basados en Registros de Desplazamiento con Realimentación Lineal (Linear Feedback Shift Registers) LFSRs [4], cuyas secuencias de salida, las *PN*-secuencias, se combinan entre sí mediante algún procedimiento o función no lineal. Entre los generadores de secuencias pseudoaleatorias más conocidos y mejor estudiados con aplicaciones criptográficas podemos señalar: generadores combinacionales, filtros no lineales, generadores controlados por uno o varios relojes, generadores multi-velocidad, generadores con decimación irregular etc. Todas estas estructuras producen a su salida secuencias cifrantes con una alta complejidad lineal, períodos muy largos y buenas propiedades estadísticas [5], [6].

Por otro lado, se ha demostrado [7] que estas mismas secuencias pseudoaletorias (*PN*-secuencias) obtenidas a partir de LFSRs con polinomios primitivos [4] pueden también generarse mediante otras estructuras matemáticas como son los Autómatas Celulares. En este sentido los Autómatas Celulares se consideran como una alternativa a los LFSRs para la generación de secuencias cifrantes en el procedimiento de cifrado en flujo. Sin embargo, la importancia real de los Autómatas Celulares radica en que algunos generadores criptográficos basados en LFSRs y diseñados como estructuras no lineales pueden modelizarse por medio de estructuras lineales basadas en Autómatas. Así pues la característica más sobresaliente de esta modelización es que muchas secuencias de uso criptográfico pueden generarse a partir de un modelo lineal basado en Autómatas. Hemos logrado pues un modelo que es lineal para una estructura que no lo es. Esto conlleva grandes ventajas a la hora del análisis de estos generadores de secuencia y, por supuesto, desde el punto de vista de su criptoanálisis.

En este trabajo se presenta el caso de un generador de secuencia cifrante como es el generador Auto-Shrinking [8] cuyo funcionamiento se basa en la decimación irregular de una *PN*-secuencia y que, sin embargo, puede modelizarse linealmente mediante Autómatas Celulares.

El procedimiento de linealización es muy sencillo y el generador, una vez que ha sido expresado en términos de Autómatas, puede implementarse fácilmente en Hardware con puertas lógicas o mediante FPGAs (Fast Programmable Gate Array logic).

Este trabajo ha sido realizado en el marco del proyecto "HESPERIA" <http://www.proyecto-hesperia.org> financiado por el Centro para el Desarrollo Tecnológico Industrial (CDTI) a través del programa CENIT y por las empresas: Soluziona Consultoría y Tecnología, Unión Fenosa, Tecnobit, Visual-Tools, BrainStorm, SAC y TechnoSafe.

[1]Física Aplicada del CSIC, C/ Serrano 144, 28006 Madrid. amparo@iec.csic.es.

[2]ITBA Instituto Tecnológico de Buenos Aires, Av. E. Madero 399 (C1106ACD), Buenos Aires, Argentina. eugepazorobles@gmail.com.

[3]DEIOC de la Facultad de Matemáticas de la Universidad de La Laguna, 38271 Tenerife, Islas Canarias. pcaballe@ull.es.

El trabajo aparece organizado de la siguiente manera: en la Sección 2 se presentan las dos estructuras básicas que vamos a utilizar (Generador Auto-Shrinking y Autómatas Celulares). En la Sección 3 se describe el procedimiento de linealización del generador Auto-Shrinking en términos de Autómatas lineales. La implementación hardware de este modelo ya linealizado aparece desarrollada en la Sección 4 y, por último, las conclusiones enunciadas en la Sección 5 completan el trabajo.

II. Fundamentos y estructuras básicas

En esta Sección se presentan algunas propiedades fundamentales de las dos estructuras básicas que se analizan en el trabajo: el Generador Auto-Shrinking y un tipo particular de Autómata Celular lineal híbrido con contenido binario.

A. El Generador Auto-Shrinking

El generador Auto-Shrinking fue diseñado por Meier y Staffelbach [8] para su uso en aplicaciones criptográficas. Más concretamente, en aquellas aplicaciones que por razones de velocidad necesitan hacer uso de cifrado en flujo. Este generador es de fácil implementación y consiste en un único LFSR de L etapas y polinomio de realimentación primitivo [4]. Este registro genera una única secuencia pseudoaleatoria (*PN*-secuencia), notada $\{a_n\}$, la cual es decimada de forma irregular dando origen a la secuencia *auto-shrunken*, notada $\{z_n\}$, o secuencia de salida del generador. La regla de decimación es extremadamente simple:

Se consideran pares $(a_{2i}, a_{2i+1})\,(i = 0,1,2,\ldots)$ de bits consecutivos no solapados de $\{a_n\}$ tales que:

Si $a_{2i} = 1$, entonces $z_j = a_{2i+1}$.

Si $a_{2i} = 0$, entonces a_{2i+1} se descarta.

Es decir, si el primer bit del par considerado es un 1, entonces el segundo bit se inserta en la secuencia de salida. Por el contrario, si el primer bit del par considerado es un 0, entonces el segundo bit se rechaza. De esta manera, se van eliminando determinados bits de la secuencia $\{a_n\}$ y los que quedan constituyen la secuencia $\{z_n\}$ o secuencia auto-shrunken. La clave de este generador es el estado inicial del LFSR aunque los autores recomiendan que el polinomio de realimentación forme también parte de ella. De acuerdo con [8], el período, complejidad lineal y propiedades estadísticas de la secuencia $\{z_n\}$ son muy adecuadas para su aplicación en criptografía.

B. Autómatas Celulares

Los Autómatas Celulares son estructuras algebraicas [9] con un número de estados finitos que se emplean en aplicaciones tan diversas como pueden ser: simulación de sistemas físicos, procesos biológicos, evolución de especies, modelos socio-económicos o criptografía. Se definen como arrays de celdas idénticas en un espacio n-dimensional y están caracterizados por los siguientes parámetros: su geometría celular, su especificación de vecindad, su número de contenidos por celda y sus reglas de transición para calcular el siguiente estado.

En este trabajo, vamos a centrarnos en un tipo particular de Autómatas mono-dimensionales, cuya vecindad abarca tan solo tres celdas, con contenido binario y cuyas reglas de transición son sencillas y lineales. En efecto, vamos a considerar Autómatas Celulares constituidos por L celdas interconectadas, cuyo contenido binario evoluciona en el tiempo de acuerdo con unas reglas de transición específicas definidas por:

- *regla 90* → $x_{t+1}^{i} = x_t^{i-1} \oplus x_t^{i+1}$
- *regla 150* → $x_{t+1}^{i} = x_t^{i-1} \oplus x_t^{i} \oplus x_t^{i+1}$

Donde x_{t+1}^{i} representa el contenido de la i-ésima celda en el instante $t+1$ para $(i = 1, \ldots, L)$ y el símbolo $\oplus$ representa la operación lógica XOR. Los Autómatas considerados son *híbridos* (diferentes celdas obedecen a diferentes reglas) y de *extremos nulos* (las celdas adyacentes a las celdas extremas están permanentemente conectadas a 0). Hay que hacer constar que ambas reglas son lineales y que incluyen tan solo la adición de dos bits (regla 90) o tres bits (regla 150). Las Tablas I y II describen el comportamiento de las reglas 90 y 150 respectivamente, para las ocho posibles configuraciones binarias de tres bits.

TABLA I
REGLA 90 PARA LAS 8 POSIBLES CONFIGURACIONES BINARIAS DE 3 BITS

111	110	101	100	011	010	001	000
0	1	0	1	1	0	1	0

TABLA II
REGLA 150 PARA LAS 8 POSIBLES CONFIGURACIONES BINARIAS DE 3 BITS

111	110	101	100	011	010	001	000
1	0	0	1	0	1	1	0

Se denomina *estado* del Autómata al contenido binario de las L celdas en cada instante de tiempo t. Para un Autómata Celular de longitud $L=6$ celdas, reglas de transición (90, 150, 90, 150, 150, 90) y estado inicial (0,0,0,1,1,1), la Tabla III describe el comportamiento de esta estructura: La formación de sus secuencias de salida (secuencias binarias leídas en vertical) así como la sucesión de estados (configuraciones binarias de 6 bits leídas en horizontal). Cabe mencionar que todas las secuencias de salida en un ciclo de estados, poseen el mismo período, la misma complejidad lineal y el mismo polinomio característico.

TABLA III
AUTOMATA CELULAR LINEAL 90/150 CON 6 CELDAS

90	150	90	150	150	90
0	0	0	1	1	1
0	0	1	0	1	1
0	1	0	0	0	1
1	1	1	0	1	0

Una forma natural de representar este tipo de Autómata es una L-tupla (*vector de regla*) notado $\Delta_L = (d_1, \ldots, d_L)$ donde $d_i = 0$ si la columna i-ésima corresponde a la regla 90 mientras que $d_i = 1$ si la columna i-ésima corresponde a la regla 150.

Las Fig. 1 y 2 representan la sucesión de los primeros 50 estados para un Autómata uniforme de regla única 90 y regla única 150 respectivamente, empezando en ambos casos en el estado inicial todo ceros salvo un 1 en la celda central.

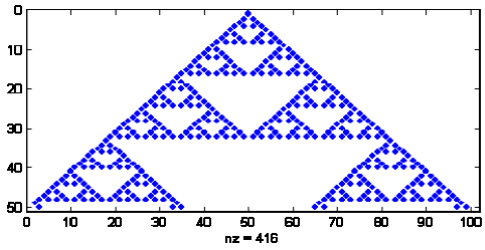

Fig. 1. Sucesión de 50 estados de un Autómata uniforme con regla 90

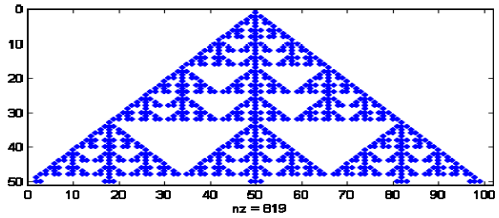

Fig. 2. Sucesión de 50 estados de un Autómata uniforme con regla 150

El polinomio característico $P_L(x)$ del autómata con L celdas se obtiene fácilmente a partir de su vector de regla como $P_L(x) = (x + d_1)(x + d_2)\ldots(x + d_L)$. Al mismo tiempo $P_L(x)$ es el polinomio característico de las secuencias de salida.

III. Modelización del Generador Auto-Shrinking mediante Autómatas Celulares

Veremos que la modelización es muy sencilla de realizar y puede generalizarse a cualquier generador Auto-Shrinking, a la vez que el modelo celular es de fácil implementación.

A. Características de la secuencia Auto-Shrunken

De acuerdo con [8], el polinomio característico de la secuencia $\{z_n\}$ o secuencia producida por un generador Auto-Shrinking con LFSR de L etapas es:

$$P(x) = (x+1)^p \qquad 2^{L-2} < p \le 2^{L-1}. \tag{1}$$

Esto implica una relación de recurrencia lineal de la forma:

$$(E+1)^p \; x_n = 0 \tag{2}$$

Siendo E el operador desplazamiento que actúa sobre los términos de una secuencia, es decir $E^k x_n = x_{n+k}$. La ecuación (2) representa una ecuación en diferencias lineal con coeficientes binarios y constantes cuyo polinomio característico $P(x)$ dado por la ecuación (1) tiene una única raíz $\lambda = 1$ con multiplicidad p. Las soluciones de este tipo de ecuación son de la forma [10]:

$$x_n = \binom{n}{0} c_0 1 + \binom{n}{1} c_1 1 + \ldots + \binom{n}{p-1} c_{p-1} 1, \tag{3}$$

donde 1 representa la única raíz de $P(x)$, $c_i \in GF(2)$ son coeficientes binarios y $\binom{n}{i}$ son números combinatorios calculados módulo 2. Cada número combinatorio, al ir dando sucesivamente valores a n, define una secuencia de valores binarios con un período constante p_i. La Tabla IV muestra para diferentes números combinatorios sus correspondientes secuencias y períodos. A partir de las 2^p posibles p-tuplas de coeficientes c_i obtenemos todas las secuencias $\{x_n\}$ que satisfacen (2). De entre ellas, algunas p-tuplas de coeficientes c_i particulares generan las secuencias auto-shrunken correspondientes a LFSRs de longitud L. En resumen, las secuencias $\{x_n\}$ soluciones de (2), entre las que se encuentras las secuencias auto-shrunken, se pueden expresar como combinación lineal de unas secuencias básicas provenientes de los números combinatorios ponderadas por unos coeficientes binarios.

TABLA IV
NÚMEROS COMBINATORIOS Y SUS CORRESPONDIENTES SECUENCIAS

Nº combi.	Secuencias	p_i
$\binom{n}{0}$	1,1,1,1,1,1,1,1,1,1, …	2^0
$\binom{n}{1}$	0,1,0,1,0,1,0,1,0,1, …	2^1
$\binom{n}{2}$	0,0,1,1,0,0,1,1,0,0, …	2^2
$\binom{n}{3}$	0,0,0,1,0,0,0,1,0,0, …	2^2
$\binom{n}{4}$	0,0,0,0,1,1,1,1,0,0, …	2^3
$\binom{n}{5}$	0,0,0,0,0,1,0,1,0,0, …	2^3
$\binom{n}{6}$	0,0,0,0,0,0,1,1,0,0, …	2^3
$\binom{n}{7}$	0,0,0,0,0,0,0,1,0,0, …	2^3

B. Correspondencia LFSRs-Autómatas Celulares

Dado un polinomio primitivo (en general irreducible) $Q(x)$, el algoritmo de síntesis de Cattel y Muzzio [7]

proporciona un par de Autómatas Celulares lineales con reglas 90/150 cuyo polinomio característico es $Q(x)$.

De este modo, podemos asociar a cada LFSR de polinomio primitivo un par de Autómatas Celulares. La Tabla V muestra diversos ejemplos de esta asociación. Nótese que el par de Autómatas asociado a cada LFSR son imagen especular uno de otro.

TABLA V
ASOCIACIÓN ENTRE LFSRS Y SUS CORRESPONDIENTES AUTÓMATAS

Pol. del LFSR	Autómata 1	Autómata 2
x^4+x+1	90 150 90 150	150 90 150 90
x^4+x^3+1	150 90 150 150	150 150 90 150
x^5+x^2+1	90 150 150 150 90	90 150 150 150 90
x^5+x^3+1	90 90 150 150 90	90 150 150 90 90

La complejidad del algoritmo de síntesis anteriormente referenciado es lineal en el grado del polinomio bajo consideración. En [11] pueden encontrarse numerosos ejemplos de polinomios primitivos de grado hasta 500 y sus correspondientes Autómatas Celulares lineales basados en reglas 90/150.

A. Concatenación de Autómatas

Ya que el polinomio característico del generador Auto-Shrinking (1) es un único factor multiplicado por sí mismo un número de veces, parece bastante natural construir su correspondiente Autómata por concatenación del autómata asociado al factor que se repite. El siguiente resultado es una formalización de esta idea.

Teorema 1. Sea C un Autómata Celular 90/150 de longitud L con $\Delta_L=(d_1,\ldots,d_L)$ y polinomio característico $P(x)$. Sea C' el Autómata imagen especular del anterior con $\Delta'_L=(d_L,\ldots,d_1)$ y la misma longitud y polinomio característico que C. Entonces $\Delta_{2L}=(d_1,\ldots,\overline{d}_L,\overline{d}_L,\ldots,d_1)$ representa un Autómata Celular 90/150 de longitud $2L$ con polinomio característico $P(x)^2$.

La demostración de este Teorema [12] está basada en la relación de recurrencia lineal del polinomio característico de los sucesivos Sub-autómatas de un Autómata dado. El resultado puede iterarse para sucesivos polinomios y vectores de regla.

$$P(x)\rightarrow\Delta_L=(d_1,d_2,\ldots,d_L)$$
$$P(x)^2\rightarrow\Delta_{2L}=(d_1,d_2,\ldots,\overline{d}_L,\overline{d}_L,\ldots,d_2,d_1)$$
$$P(x)^{2^2}\rightarrow\Delta_{2^2L}=(d_1,\ldots,\overline{d}_L,\overline{d}_L,\ldots,\overline{d}_1,\overline{d}_1,\ldots,\overline{d}_L,\overline{d}_L,\ldots,d_1)$$

Nótese que el Autómata básico se concatena con su versión especular tras la complementación de la última regla (regla que ocupa la posición menos significativa). Sucesivas aplicaciones de este resultado nos proporcionan Autómatas celulares de polinomios característicos:

$$P(x),P(x)^2,P(x)^{2^2},P(x)^{2^3},\ldots,P(x)^{2^q}$$

y de longitudes $L, 2L, 2^2L, 2^3L, \ldots, 2^qL$, respectivamente.

B. Modelización del Generador Auto-Shrinking

Para la modelización del generador Auto-Shrinking en términos de Autómatas Celulares, aplicamos los resultados obtenidos en las secciones III-A, III-B y III-C.

Primeramente determinamos el Autómata Celular correspondiente al polinomio básico $x+1$. Se trata de una simple regla 150 que denotamos por $\Delta_1=(1)$. La aplicación de los resultados previos nos permite derivar las siguientes relaciones entre polinomios y vectores regla:

$$\begin{aligned}
&(x+1)\rightarrow\Delta_1=(1)\\
&(x+1)^2\rightarrow\Delta_2=(0,0)\\
&(x+1)^4\rightarrow\Delta_4=(0,1,1,0)\\
&\quad\vdots\qquad\qquad\vdots\\
&(x+1)^{2^{L-1}}\rightarrow\Delta_{2^{L-1}}=(0,1,1,\ldots,1,1,0)
\end{aligned}$$

De esta manera, se han ido obteniendo los vectores regla mediante una fácil asociación entre polinomios de la forma $(x+1)^{2^q}$ y sus correspondientes Autómatas Celulares Δ_{2^q}. Puede observarse que la forma de estos Autómatas es muy simple: tienen una longitud 2^q con reglas 90 en los extremos y reglas 150 en las restantes celdas.

De acuerdo con (1) y ya que el parámetro p está acotado por $2^{L-2}<p\le 2^{L-1}$, el último vector $\Delta_{2^{L-1}}$ corresponde a la representación del Autómata que genera la correspondiente secuencia auto-shrunken.

C. Un ejemplo ilustrativo

Sea $\{z_n\}$=$\{0,0,0,1,1,1,1,0\}$ la secuencia auto-shrunken generada por un LFSR de $L=4$ etapas, polinomio de realimentación dado por x^4+x+1 y estado inicial $(1,0,0,0)$. La secuencia $\{z_n\}$ tiene periodo $T=8$, complejidad lineal $LC=5$ y polinomio característico $P(x)=(x+1)^5$.

De acuerdo con los resultados anteriores el Autómata Celular 90/150 que genera esta secuencia es:

$$\Delta_8=(0,1,1,1,1,1,1,0)$$

En la Tabla VI se aprecia el Autómata y las secuencia que se van generando durante la evolución del mismo empezando en el estado inicial $(0,0,0,1,0,1,1,1)$. Vemos que en la columna del extremo izquierdo (regla 90) se genera la secuencia $\{z_n\}$. Nótese que la columna adyacente (regla 150) genera la misma secuencia pero desplazadas una posición hacia arriba.

Debe destacarse también la simetría de estos Autómatas, donde se aprecia que en las columnas simétricas entre sí se van obteniendo las mismas secuencias pero desplazadas $T/2$ posiciones.

D. Consideraciones generales sobre la modelización de los generadores Auto-Shrinking

A continuación se exponen algunas consideraciones con respecto a la modelización de estos Generadores:

1) La regla que los caracteriza tiene una estructura simétrica: regla 90 en las celdas extremas y regla 150 en las celdas interiores.

2) El Autómata $\Delta_{2^{L-1}}$ genera todas las secuencias que son soluciones de la ecuación en diferencias:

$$(E+1)^{2^{L-1}}\ x_n = 0. \tag{4}$$

Esto es

$$x_n = \sum_{i=0}^{2^{L-1}-1} \binom{n}{i} c_i 1. \tag{5}$$

Por tanto, las secuencias auto-shrunken se obtienen como soluciones particulares de (5) con $c_i = 0 \quad \forall i \geq p$.

TABLA VI
UN AUTÓMATA 90/150 GENERANDO UNA SECUENCIA AUTO-SHRUNKEN

90	150	150	150	150	150	150	90
0	0	0	1	0	1	1	1
0	0	1	1	0	0	1	1
0	1	0	0	1	1	0	1
1	1	1	1	0	0	0	0
1	1	1	0	1	0	0	0
1	1	0	0	1	1	0	0
1	0	1	1	0	0	1	0
0	0	0	0	1	1	1	1

3) El Autómata $\Delta_{2^{L-1}}$ genera todas las secuencias auto-shrunken producidas por todos los LFSRs de polinomio primitivo y *L* etapas. En este caso, el conocimiento del registro de desplazamiento (polinomio de realimentación) sobre el que se basa el generador Auto-Shrinking no es necesario. Es decir hay una parte del generador considerada como clave que para nuestra modelización mediante Autómatas resulta irrelevante.

4) El Autómata $\Delta_{2^{L-1}}$ genera todas las secuencias auto-shrunken correspondientes a LFSRs de polinomio primitivo y número de etapas < L. Esto significa que el Autómata más largo incluye todas las secuencias correspondientes a los Autómatas de menor tamaño empezando en un estado inicial simétrico.

5) La modelización de estas estructuras lineales con reglas 90/150 es muy adecuada para su implementación con lógica FPGA. Esta característica los hace muy convenientes en desarrollos donde la velocidad es relevante, como puede ser en sistemas de cifrado en flujo o en sistemas de comunicaciones con altas tasas de transmisión.

6) La linealidad de estos modelos celulares así como las simetrías encontradas, véase la Tabla VI, resultan muy ventajosas para la reconstrucción de la secuencia cifrante a partir de porciones de secuencia interceptada.

IV. LÓGICA PROGRAMABLE

A **Field Programmable Gate Array** (FPGA) es un semiconductor que contiene componentes con lógica programable e interconexiones programables. Los componentes lógicos programables pueden ser programados para duplicar la funcionalidad de compuertas lógicas básicas como son la AND, OR, XOR, NOT o incluso funciones combinacionales más complejas, tales como decodificadores o funciones matemáticas. En la mayoría de las FPGAs, los components lógicos programables, los bloques lógicos, incluyen también los elementos de memoria. Dichos elementos pueden ser simples Flip-Flops o bloques de memoria más completos.

A. Aplicaciones

Las FPGAs básicamente encuentran aplicación en áreas donde sea de especial interés el uso del procesamiento en paralelo que esta arquitectura puede ofrecer. También son aplicables en desarrollos donde sea importante la velocidad. Es muy fácil el diseño de un circuito y los cambios que él mismo requiera dado que este tipo de chips es precisamente muy versátil.

B. Arquitectura

La arquitectura básica consiste en un arreglo de bloques lógicos programables (CLBs) y canales de ruteo. Un circuito a desarrollar puede ser mapeado en una FPGA con la programación adecuada.

Un bloque típico de una FPGA consiste de una tabla (look up table) de 4 entradas (LUT) y un flip-flop como se muestra a continuación en la Fig. 3.

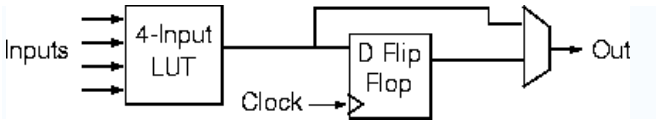

Fig. 3. Bloque Lógico

Hay sólo una salida, que puede ser directamente la salida de la LUT pero que primero entra en el flip-flop. Las entradas al bloque lógico son las 4 entradas a la LUT y la entrada del clock.

En el caso de Xilinx [13] y para la famila Spartan de FPGAs, los chips ofrecen del orden de 50.000 a 1,4 millones de sistemas de compuertas con lo cual la capacidad de desarrollar un circuito con estos componentes es muy grande. Su rapidez es similar a la de los ASICs, pero, a diferencia de aquellos, estos chips aportan la ventaja de su versatilidad. De hecho las FPGAs no tienen los altos costos iniciales y su programabilidad permite realizar cambios y upgrades en el hardware del circuito diseñado.

C. Programación de un Registro de Desplazamiento

El generador Auto Shirnking se basa en un Registro de desplazamiento con realimentación lineal o LFSR.

La implementación de un registro de desplazamiento mediante FPGAs es muy sencilla y puede llevarse a cabo utilizando configuraciones ya disponibles en los chips FPGA. En cada ciclo de reloj, la entrada es introducida en la entrada DIN, como vemos en la Fig. 4. En cada ciclo de reloj, el bit de entrada es desplazado al siguiente flip-flop.

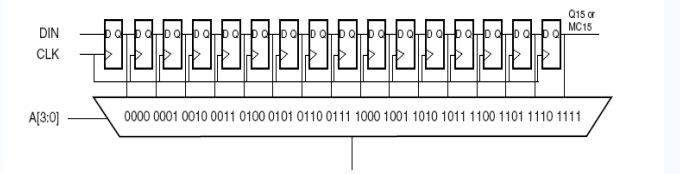

Fig. 4. LUT configurado como un Registro de Desplazamiento

D. Registros de Desplazamiento con Realimentación Lineal (LFSR)

Los LFSRs con polinomio de realimentación primitivo necesitan conexiones tomadas de posiciones específicas dentro del registro. El LFSR pueder ser realizado a partir de la configuración SRL16, disponible en las FPGAs. Una forma de diseño es direccionar el bit necesario de la configuración SRL16 mientras se conecta el Q15 en cascada a la siguiente configuración SRL16. Por ejemplo, la Fig. 5 muestra una implementación de un LFSR de 52 etapas con realimentación en las etapas 49 y 52 [15]. Vemos que los bits de las etapas realimentadas entran a la primera posición como bit 1 en el siguiente clock.

Lo anteriormente explicado sirve para ejemplificar un LFSR de más de 16 etapas y es una aplicación de cómo se colocan en cascada las configuraciones que ya están incluidas en estos tipos de chips como pueden ser la SRL16 que nos ocupa.

Como vemos, esta configuración consiste en colocar en cascada una serie de flip flops, es decir que también es posible, mediante el uso de una única configuración tipo flip flop, armar un LFSR con tantos flip flops como etapas tiene el LFSR. Las consideraciones que hay que tener presente son las realimentaciones a la salida del flip flop correspondiente.
En la Tabla VII podemos apreciar las realimentaciones necesarias para la configuración de LFSRs máximos de n etapas [14].

E. Autómatas Celulares

Como vimos para las reglas 90 y 150, el contenido de la celda en el instante $t+1$ se obtiene realizando una XOR entre el contenido de ya sean dos o tres celdas, respectivamente, en el instante *t*.

Para programar esta topología en una FPGA bastará con asignar un lugar de memoria dentro del chip, en el caso de FPGAs de la familia Spartan considerada, este lugar de memoria podrá albergar el contenido de hasta 512x36 bits en cada bloque, pudiéndose superponer varios de éstos.

Para un autómata que modela un generador Auto-Shrinking de $L=4$ etapas tendríamos un vector regla del tipo $\Delta_8=(0,1,1,1,1,1,1,0)$ como vimos en la sección III-E.

Este autómata lo podríamos programar en Hardware rápidamente, véase la Fig. 6, asignando un lugar de memoria con 8 bits y asociando estos bits a compuertas lógicas XOR para efectuar la regla que corresponda.

TABLA VII
LISTA DE LAS CONEXIONES NECESARIAS PARA LFSRS MÁXIMOS

Linear Feedback Shift Registers in Virtex Devices XILINX®

Table 1: **Taps for Maximum-Length LFSR Counters**

n	XNOR from	n	XNOR from	n	XNOR from	n	XNOR from
28	28,25	70	70,69,55,54	112	112,110,69,67	154	154,152,27,25
29	29,27	71	71,65	113	113,104	155	155,154,124,123
30	30,6,4,1	72	72,66,25,19	114	114,113,33,32	156	156,155,41,40
31	31,28	73	73,48	115	115,114,101,100	157	157,156,131,130
32	32,22,2,1	74	74,73,59,58	116	116,115,46,45	158	158,157,132,131
33	33,20	75	75,74,65,64	117	117,115,99,97	159	159,128
34	34,27,2,1	76	76,75,41,40	118	118,85	160	160,159,142,141
35	35,33	77	77,76,47,46	119	119,111	161	161,143
36	36,25	78	78,77,59,58	120	120,113,9,2	162	162,161,75,74
37	37,5,4,3,2,1	79	79,70	121	121,103	163	163,162,104,103
38	38,6,5,1	80	80,79,43,42	122	122,121,63,62	164	164,163,151,150
39	39,35	81	81,77	123	123,121	165	165,164,135,134
40	40,38,21,19	82	82,79,47,44	124	124,87	166	166,165,128,127
41	41,38	83	83,82,38,37	125	125,124,18,17	167	167,161
42	42,41,20,19	84	84,71	126	126,125,90,89	168	168,166,153,151
43	43,42,38,37	85	85,84,58,57	127	127,126		
44	44,43,18,17	86	86,85,74,73	128	128,126,101,99		

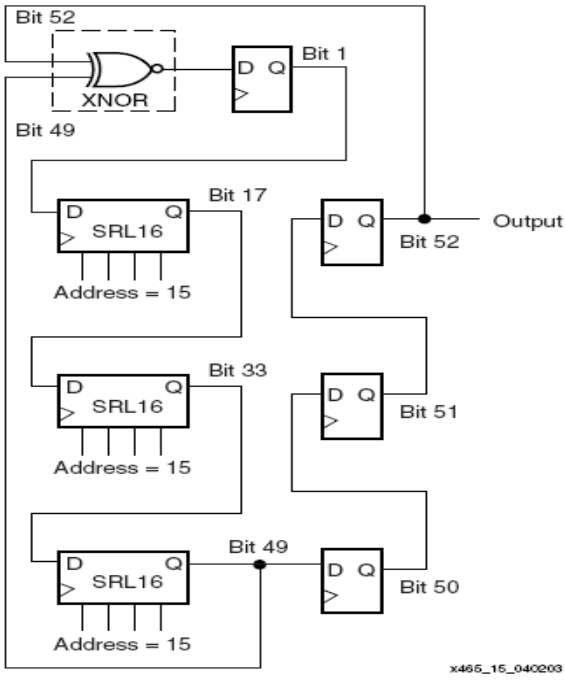

Fig. 5. LFSR de 52 etapas

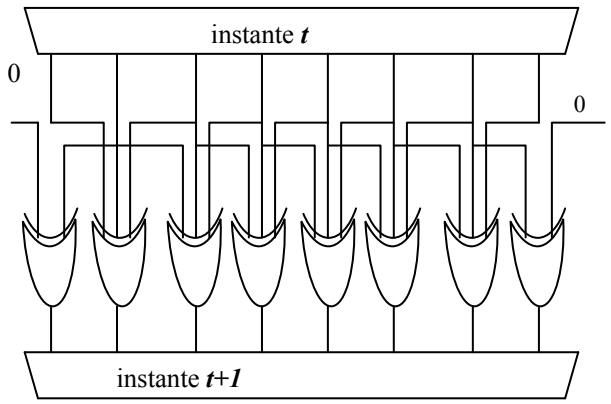

Fig. 6. Diseño de circuito lógico para Autómata Δ_8=(0,1,1,1,1,1,1,0)

V. CONCLUSIONES

Las secuencias Auto-shrunken son soluciones particulares de un tipo de ecuaciones en diferencias lineales y pueden ser generadas a partir de Autómatas Celulares lineales basados en reglas 90/150. El modelo celular es muy sencillo y presenta además de la linealidad unas propiedades de simetría que pueden ser convenientemente utilizadas para la reconstrucción de la secuencia cifrante a partir de porciones de secuencia interceptada. Al mismo tiempo, se han analizado las propiedades estructurales de las secuencias auto-shrunken que son una simple combinación lineal de unas secuencias tipo. De esta manera, vemos que una familia de generadores de secuencia de uso criptográfico que han sido concebidos y diseñados a partir de una estructura no lineal basada en LFSRs pueden ser linealizadas en términos de un modelo simple basado en Autómatas Celulares. El procedimiento de linealización es inmediato y puede ser implementado para aplicaciones criptográficas actuales y concretas. La forma particular de estos Autómatas se presta a una implementación hardware con lógica FPGA.

AGRADECIMIENTOS

Este trabajo se realiza en el marco de una Tesis Doctoral, por ende los autores quieren expresar su agradecimiento al ITBA, Instituto Tecnológico de Buenos Aires (Argentina) y a alguno de sus profesores por sus útiles comentarios durante la preparación de este manuscrito.

REFERENCIAS

[1] GSM, Global Systems for Mobile Communications, available at http://cryptome.org/gsm-a512.htm
[2] Bluetooth, Specifications of the Bluetooth system, Version 1.1, 2001, available at http://www.bluetooth.com/
[3] R.L. Rivest, RSA Data Security, Inc., March 12, 1998.
[4] S. Golomb, *Shift-Register Sequences,* Aegean Press, New York, 1982.
[5] P. Caballero-Gil, and A. Fúster-Sabater, "A Wide Family of Nonlinear Filter Functions with a Large Linear Span," *Information Sciences*, vol.164, no. 4, pp. 197-207, Aug. 2004.
[6] A. Fúster-Sabater, "Run Distribution in Nonlinear Binary Generators," Applied Mathematics Letters, vol. 17, no. 12, pp. 1427-1432, Dec. 2004.
[7] K. Cattell, and J.C. Muzio. "Synthesis of One-Dimensional Linear Hybrid Cellular Automata," IEEE Trans. Computers-Aided Design, vol. 15, no. 3, pp. 325-335, 1996.
[8] W. Meier, and O. Staffelbach. "The Self-Shrinking Generator," *Proc. Eurocrypt* '94, LNCS, Springer-Verlag, vol. 950, pp. 205-214, 1994.
[9] J. Kari, "Theory of cellular automata: A survey," *Theoretical Computer Science*, vol. 334, no. 3, pp. 3-33, Sep. 2005.
[10] R. Lidl, and H. Niederreiter, *Introduction to Finite Fields and Their Applications*, Cambridge University Press, Cambridge, England, 1986.
[11] K. Cattell, and Z. Shujian. "Minimal Cost One-Dimensional Linear Hybrid Cellular Automata of Degree through 500," *J of Electronic Testing: Theory and Applications*, vol. 6, no. 3, pp. 255-258, 1995.
[12] A. Fúster-Sabater, and P. Caballero-Gil, "Automata in Cryptanalysis of Stream Ciphers," *Proc. ACRI* '06, LNCS, Springer-Verlag, vol. 4176, pp. 611-616, 2006.
[13] Xilinx, www.Xilinx.com. Spartan 3E Product Brief, pn0010855.pdf
[14] Xilinx, XAPP052.pdf white paper
[15] Xilinx, XAPP210.pdf white paper